Adaptive Sampling of Motion Trajectories for Discrete Task-bases Analysis and Synthesis of Gesture

Pierre-François Marteau, Sylvie Gibet

Adaptive Sampling of Motion Trajectories for Discrete Task-Based Analysis and Synthesis of Gesture

Pierre-François Marteau and Sylvie Gibet

Valoria, Université de Bretagne Sud, Campus de Tohannic, rue Yves Mainguy, F-56000 Vannes, France
{Pierre-Francois.Marteau, Sylvie.Gibet}@univ-ubs.fr

Abstract. This paper addresses the problem of synthesizing in real time the motion of realistic virtual characters with a physics-based model from the analysis of human motion data. The synthesis is achieved by computing the motion equations of a dynamical model controlled by a sensory motor feedback loop with a non-parametric learning approach. The analysis is directly applied on end-effector trajectories captured from human motion. We have developed a Dynamic Programming Piecewise Linear Approximation model (*DPPLA*) that generates the discretization of these 3D Cartesian trajectories. The *DPPLA* algorithm leads to the identification of discrete target-patterns that constitute an adaptive sampling of the initial end-point trajectory. These sequences of samples non uniformly distributed along the trajectory are used as input of our sensory motor system. The synthesis of motion is illustrated on a dynamical model of a hand-arm system, each arm being represented by seven degrees of freedom. We show that the algorithm works on multi-dimensional variables and reduces the information flow at the command level with a good compression rate, thus providing a technique for motion data indexing and retrieval. Furthermore, the adaptive sampling seems to be correlated with some invariant law of human motion.

1 Introduction

When simulating and animating virtual characters, biologically-inspired models play a major role, as the produced movements exhibit properties inherent to human movements. As emphasized by psychologists and physiologists, human beings are more sensitive to movement of biological origin [1]. There are two ways to reach a certain degree of naturalness for the synthesis of gestures. First, the articulated system can be modeled by a physical model responding to physical laws of movement. The difficulty here is not so much to simulate the motion equations, but to determine the appropriate controller that drives the system towards a desired goal expressed in the task-command space. Second, the animation can be directly done by motion capture data. The difficulty with this last method is to determine new motion on the basis of previously registered elementary motions, and to ensure smooth transitions between these elementary motions.

In our approach, we try to establish a link between real-time synthesis models and motion analysis models using motion capture data. The synthesis model is based on a dynamical Sensory Motor Model (*SMM*) and is driven by a task-based analysis model that extracts discrete target-based patterns on the basis of motion capture data.

S. Gibet, N. Courty, and J.-F. Kamp (Eds.): GW 2005, LNAI 3881, pp. 224 – 235, 2006.

After presenting the global analysis synthesis method, this paper is focused on the presentation of the analysis model. More precisely we describe a Dynamic Programming Piecewise Linear Approximation (*DPPLA*) algorithm whose outputs can be used to control the articulated system. Since the command is directly extracted from real human data, the animation system produces movement with biological relevance. The analysis-synthesis model is applied to the simulation of anthropomorphic hand-arm movements. The performance of the method is presented according to invariant laws of movement.

2 Analysis-Synthesis Methods

Several problems arise when simulating human hand-arm motion. First, the biomechanical system composed of a set of interacting articulated structures has to be has to be modeled. This biomechanical model is directly dependent on the way the muscular-skeleton apparatus is modeled. Most of the time simplifications of the mechanical structure have to be considered. Controlling such a complex system necessitates the design of appropriate controllers associated to the different articulated chains. In our approach, the control of the biomechanical system is materialized by a Sensory Motor Model (*SMM*) that continuously uses sensory data to update the state variables of the dynamical system to control. The feedback mechanism carries out an inversion process, i.e. it automatically computes the input of the biomechanical system from the observable outputs and the command input. The *SMM* is illustrated in Figure 1, as the *motion synthesis* block.

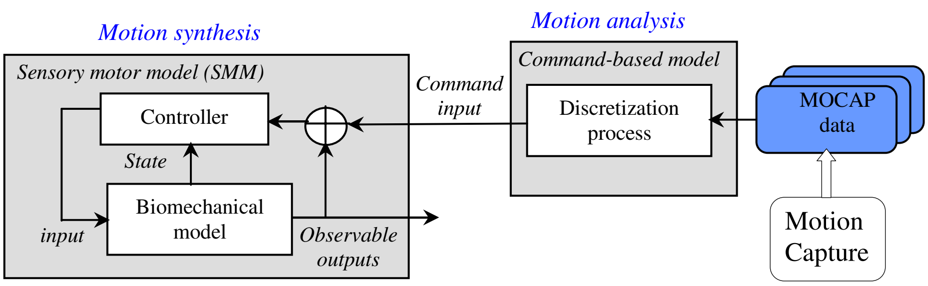

Fig. 1. Analysis/synthesis model for gesture modeling and animation

This paper deals with the problem of modeling the command of such a sensory motor model, as a *motion analysis* approach. This analysis process can be seen as an inversion process: from motion capture signals, a discrete command pattern is extracted through a discretization process; the command input can be used to control the *SMM*, as illustrated in Figure 1.

2.1 Motion Analysis

Exploiting analysis data to run the synthesis model raises the question of identifying the appropriate variables at the command level. Some psychologists and physiologists assume that - at least for some classes of movements - spatial representation is more invariant than force-time patterns or joint rotations. Following this idea, we make the hypothesis that end-effectors motion traces expressed in the 3D Cartesian space can be used to control the muscular-skeleton system. We expect that this spatial representation is closer to the task than other internal sensory or motor variables, as force or moment variables [2-3].

More precisely, this paper will cover the modeling of the end-effectors trajectories in terms of discrete target patterns. Given a desired trajectory, we extract sequences of targets that represent in an optimal way the original trajectory. This approach differs from the methods widely used for the purpose of segmenting human motion capture data into high-level behaviors or low-level components. These last methods are designed most of the time to reduce dimensionality by identifying low-dimensional clusters in high-dimensional data [4]. Another method is proposed for non-uniform sub-sampling of motion captured data. This method uses polygonal approximation to provide a compressed representation of dance gesture trajectories [5]. The objectives of this approach are different from ours, since the compressed data in [5] are used as input of a recognition system, whereas we use our reduced trajectories for synthesis purpose.

Our target-based discretization method, also called *adaptive sampling* could be useful for motion segmentation, but above all we aim to extract discrete patterns as input of our motion generation models. The main interest of the adaptive sampling is to facilitate the manipulation (edition, recombination) of elementary movements, the composition operators being implemented as the concatenation and smoothing of target sequences. Furthermore this discretization process leads to the reduction of the data flow at the command level and to the reduction of the representation space in which movements are embedded. This is also of great importance for dealing with information retrieval in movement data base. Finally this sampling process is a first step towards the parameterization of motion. Rather than using straightforward key-points extracted from motion invariant laws [6], we propose an automatic segmentation process operating on multi-channel variables which yields a discrete representation of motion.

2.2 Motion Synthesis

Numerous solutions exist to control sensory motor systems. Some methods consider the control problem as finding numerical solutions to inverse kinematics or inverse dynamics, depending on the representation of the movement system. Among these solutions, we have developed analytical methods extended by learning methods, applied both for kinematics or dynamics control.

The synthesis model attached to each articulated chain is represented by a sensory motor system which yields a means of coupling the continuous internal signals in the sensory motor process with desired commands expressed as continuous trajectories in the Cartesian space or as sequences of discrete targets in the task space. It has already been developed for controlling various articulated systems with different control policies: The *GSM* model uses a gradient-based algorithm in a sensory motor closed-loop transformation which integrates neurophysiological elements [7]. This model has proved to control articulated chains and produce motion that globally respects human motion laws. It has been used in a modular architecture to generate expressive communicative and Sign Language Gestures [8-9] or coordinated juggling motion [10].

Another control policy uses a learning approach within a sensory motor loop (*ASM* model) [11-12]. In this case, the learning algorithm is based on a local inversion principle and uses local neighborhood research techniques to compute the new predicted state variation to be generated. The training data set can be incrementally updated in time to adapt to changes of the performance tasks and to structural changes of the physical system.

A sensory motor dynamical system is considered in this paper, which includes two inversion processes: the first one is a kinematics inversion, based on the same learning control policy; the second one is a dynamics inversion, achieved by a Proportional Integrative Derivative model.

For each sensory motor model, various tasks can be defined, such as reaching or tracking tasks. In our synthesis models, the task is expressed as desired goals in the spatial 3D space or in the joint angular space. For simple or multiple reaching tasks, these goals can be represented as sequences of targets to reach, with or without co-articulation. For tracking tasks, the goals can be expressed as desired end-point continuous or discrete trajectories.

We propose to define the input command from the analysis of motion capture data. Coupling synthesis with data extracted from human motion is necessary if we want to integrate some invariant features of movements. The discretization method, called Dynamic Programming Piecewise Linear Approximation (*DPPLA*), achieves the analysis process necessary for the synthesis process, which was missing in our former systems.

3 Dynamic Programming Piecewise Linear Approximation Model (*DPPLA*)

The *DPPLA* algorithm makes the discretization of end-effector trajectories possible in $O(n^2/k)$ where k is the number of samples. These trajectories can be considered as a multivariate continuous *3D* process $X(t) = [x(t), y(t), z(t)]$. A more general view is to consider $X(t)$ as a spatio-temporal trajectory of time-stamped spatial vectors in p dimensions. In practice, we will deal with the sampled trajectory $X(n)$ where n is the time-stamp index.

We propose a data modeling approach to handle the adaptive sampling of the end-effector trajectories. More precisely, we are seeking an approximation $X_{\hat{\theta}}$ of $X(n)$ such as:

$$\hat{\theta} = \underset{\theta}{ArgMin}\big(E(X, X_\theta)\big)$$

where E is the *RMS* error between X and the model X_θ.

As a first attempt, we have selected the family $\{ X_\theta(n) \}$ as the set of piecewise linear functions. Numerous methods have been proposed to the problem of approximating multidimensional curves using piecewise linear simplification and dynamic programming in $O(kn^2)$ complexity [13]. Some efficient algorithms [14] (in $O(nlog(n))$ complexity) [15] have been proposed for planar curves, but none for the general case in R^d. We constraint the search of the segments by imposing that the extremities of the piecewise linear segments are on the trajectory *X(t)*. Thus, θ is the set of discrete time location $\{n_i\}$ of the segments' endpoints. Since the end of a segment is the beginning of the following one, two successive segments share a common n_i at their interface. The selection of the optimal set of parameters $\hat{\theta} = \{\hat{n}_i\}$ is performed using a dynamic programming algorithm [16] as follows.

We first define the compression rate of the piecewise approximation as:

$$\rho = 1 - \frac{\left|\{n_i\}\right|}{\left|\{X(n)\}\right|} \times \frac{p+1}{p}$$

where $|A|$ stands for cardinal of set A and $X(n) \in \Re^p, \forall n$

Given a value for ρ and the size of the trajectory window to sample $w = \left|\{X(n)\}_{n \in \{1,\ldots,w\}}\right|$, the number $N = \left|\{n_i\}\right| - 1$ of piecewise linear segments is known.

Let us define $\theta(k)$ as the parameters of a piece wise approximation containing k segments, and $\delta(k,i)$ as the minimal error between the best piecewise linear approximation containing k segments and covering the discrete time window $\{1,..,i\}$:

$$\delta(k,i) = \underset{\theta(k)}{Min}\left\{\sum_{n=1}^{i} \left\| X_{\theta(k)}(n) - X(n) \right\|^2 \right\}$$

According to the Bellman optimality principle, $\delta(k,i)$ can be decomposed as follows:

$$\delta(k,i) = \underset{n_k \le i}{Min}\left\{ d(n_k, i) + \delta(k-1, n_k) \right\}$$

where $d(n_k, i) = \sum_{n=n_k}^{i} \left\| Y_{k,i}(n) - X(n) \right\|^2$ and $Y_{k,i} = (X(i) - X(n_k)).\frac{n - n_k}{i - n_k} + X(n_k)$ is the linear segment between $X(i)$ and $X(n_k)$.

The initialization of the recursion is obtained observing that:

$$\forall k, \forall i < k, \delta(k,i) = 0$$

The end of the recursion gives the optimal piecewise linear approximation, e.g. the set of discrete time locations of the extremity of the linear segments:

$$\hat{\theta}(k) = \underset{\theta(k)}{ArgMin}\left\{\sum_{n=1}^{w} \left\| X_{\theta(k)} - X(n) \right\|^2 \right\}$$

with the minimal error :

$$\delta(k,w) = \sum_{n=1}^{w} \left\| X_{\hat{\theta}(k)}(n) - X(n) \right\|^2$$

The complexity of the proposed algorithm is in $O(k.w^2)$. To reduce this complexity, the search window can be limited by using a lower bound factor for each step i: $lb = max\{i - band, 0\}$, where *band* is a parameter fixed by the user:

$$\delta(k,i) = \underset{lb \le n_k \le i}{Min}\left\{d(n_k,i) + \delta(k-1,n_k)\right\}$$

In practice we use *band = 2*w/k*, leading to the complexity $O(w^2/k)$.

4 Runtime Synthesis

Here a generic learning model for the control of a dynamical articulated model is used. The control policy associated to this learning method jointly achieves the kinematics and the dynamics inversion of the system. The kinematics inversion is carried out by a non parametric learning algorithm which processes the mapping $(y, \delta x) \rightarrow \delta y$, y being the state and x the output of the system (*ASM* model) [12]. The error signals measured between the sensory output and the task input are used as corrective information to update the torque command of the movement system.

The dynamics inversion is assured by a set of controllers acting on the pair (τ,δ*q/dt*), τ being the forces applied on the joints, and *dq/dt* the angular velocity of the joint rotations. These controllers, classically used in robotics and computer animation and issued from linear control theory use Proportional Integrative Derivative principle (*PID*). For each internal joint, each *PID* controller takes as inputs angular position of the joint and its derivative as well as the desired angular position, and computes the torque output required to produce the desired displacement of the joint as expressed by the following equation:

$$\tau(t) = K_p(\vec{q}_T - \vec{q}) + K_d(\dot{\vec{q}}_T - \dot{\vec{q}}) + K_i \int \vec{q}(t)dt$$

where q is the angle of the joint, q_T is the desired angle, K_p, K_d and K_i are the proportional, derivative and integral gains. The effect of the *PID* controller is to eliminate large step changes in the errors, thus smoothing the simulated motion.

5 Experiments

In this section, the results of two experiments are presented. One concerns the analysis process applied on motion capture sequences. The second highlights the synthesis process, using the result of the analysis process as the command input of the dynamical system. The motion data used in our experiments were captured from a VICON optical system at the rate of 110 frames/second. We recorded end-arm movements of about three s duration. Subjects were told to perform about ten different random patterns, varying the kinematics and the shape of the patterns.

The analysis is conducted on 3D Cartesian trajectories of the arm extremity. We consider for hand-arm movements that these trajectories express the trace of the task-based command. The *DPPLA* algorithm is applied on these trajectories with varying compression rates that fix the number of targets on the trajectory. The algorithm segments the trajectory by assigning cut targets along the motion sequence. The objective is not only to detect these cut-points, but also to characterize the distribution of relevant samples along the trajectory.

The results of the *DPPLA* algorithm are illustrated in Figure 2 for a compression rate of 85%. This compression rate is an optimized parameter directly linked to the number of discrete targets extracted by the *DPPLA* algorithm. Figure 2 shows the *x*, *y* and *z* curves of the end-extremity trajectories, both for the real motion capture data and for the simulated data, after applying the *DPPLA* algorithm. It also indicates the

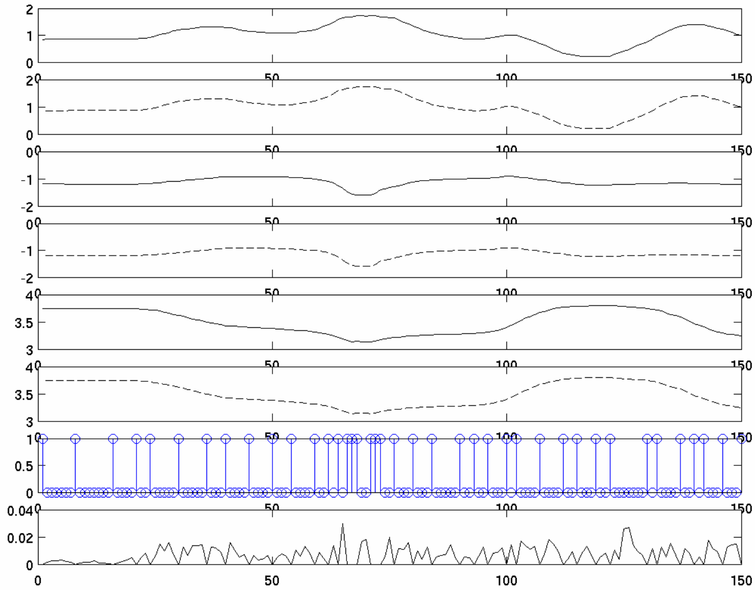

Fig. 2. Three-joint arm simulation with random pattern with a compression rate of 75%; *x, y, z* trajectories: real data *(solid)* and simulated data *(dashed)*; motion separation points assigned by the *DPPLA* algorithm. The *x*-axis corresponds to the frame number, and the vertical bars specify the target points assigned by the algorithm; Reconstruction error.

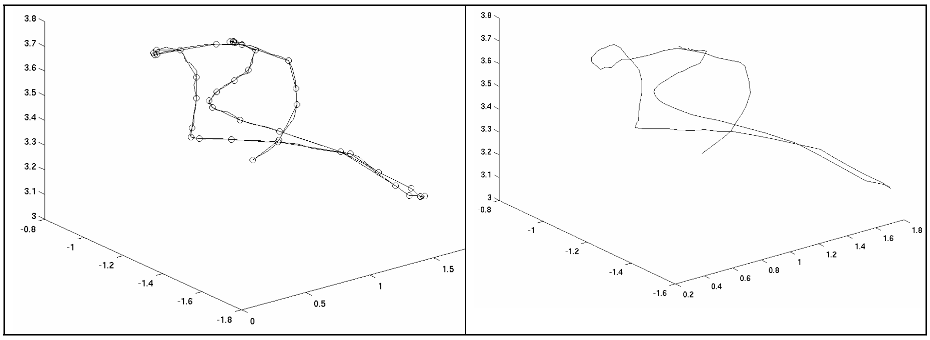

Fig. 3. *(left)* Trajectory of the human wrist in the Cartesian space with the localization of the targets: capture motion data and reconstructed data by linear interpolation; *(right)* Simulated trajectory of the wrist of the virtual dynamical humanoid

location along the frames of the cut-points (targets) that approximate the desired trajectory and the corresponding absolute error between the real data and the simulated ones.

Figure 3 *(left)* gives the shape of the real and reconstructed trajectories from the location of the targets. Some attraction areas where the targets are more concentrated can be seen in this figure: they correspond to zones of larger complexity of the signal. The segmentation induced by the *DPPLA* algorithm can be considered as measured by the density of targets along the motion.

Figure 3 *(right)* illustrates the simulated trajectory performed by the virtual character. Some overshoots can be seen when the curvature is high, due to the fact that the dynamical parameters cannot be adjusted during the course of the simulated motion.

These results demonstrate the tendency of *DPPLA* to increase the target density when the curvature increases. This correlation can be illustrated in Figure 4 with the superposition of the density function representing the spatial concentration of the targets along the motion frames, and the curvature function. Table 1 gives correlation factors between the two functions for three compression rates using a linear regression method.

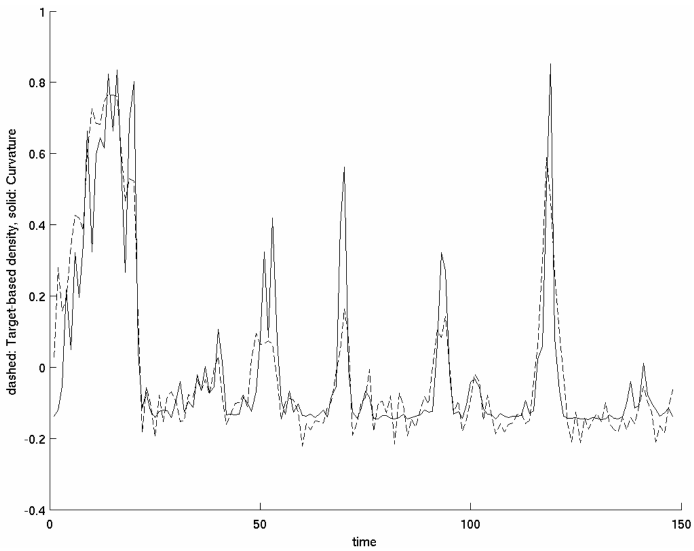

Fig. 4. *(dashed)*Target-based density evolving with time; *(solid)* Normalized curvature evolving with time. The areas where Target-density function is high correspond to areas with high curvature; this function segments the end-point trajectory.

Table 1. Correlation factors between target-based density along the trajectory and curvature for different compression rates

Compression rate	Correlation factor
65%	0.90
75%	0.88
85%	0.84

The *DPPLA* algorithm provides a way to adaptively sample the end-effector trajectory according to the variations of curvature along the trajectory. The analogy between the spatial density of targets and the curvature leads to the investigation of the density function behavior compared with the well-known "two-third power law" [17]. The latter is equivalently expressed by a *one-third power law* relating tangential velocity $v(t)$ to radius of curvature $r(t)$:

$$v(t) = k\, r(t)^{1/3}$$

This law, considered as a basic invariant characteristic of movement, has been demonstrated to be robust for 2D handwriting movements, and also for 3-D elliptical patterns. Figure 5 *(up-left)* illustrates this one-third power law for a random end-effector trajectory. Figure 5 *(up-right)* shows that a similar power relationship exists between target-based tangential velocity $v_T(t)$ and target-based density function $D_s(t)$ according to the *DPPLA* algorithm:

$$V_T(t) = K\left(\frac{1}{D_s(t)}\right)^{\gamma}$$

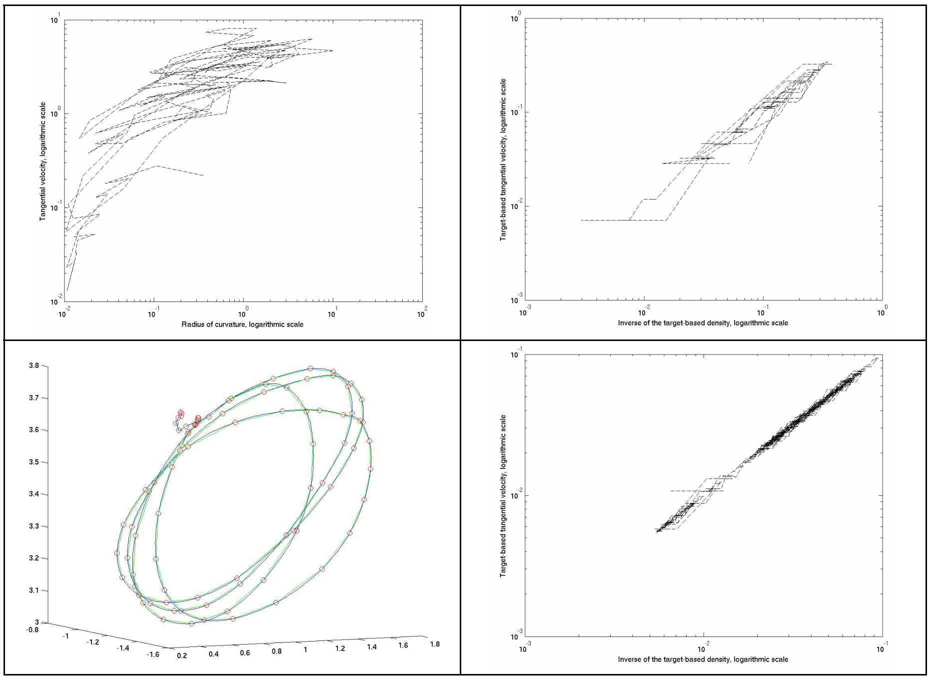

Fig. 5. *(up-left)* Tangential velocity versus radius of curvature in logarithmic scale; *(up-right)* Target-based tangential velocity versus inverse of target-based density in logarithmic scale. Both figures are traced for the complex pattern trajectory presented in Fig. 3; *(down-left)* End effector elliptical trajectory with target location; *(down-right)* Target-based tangential velocity versus inverse of target-based density in logarithmic scale.

with

$$V_T(t_i) = \frac{\|Tg_i - Tg_{i-1}\|}{t_i - t_{i-1}} = \frac{\delta Tg_i}{\delta t_i} \quad \text{et} \quad D_s(t_i) = \frac{1}{\delta Tg_i}$$

The power law between V_T and $1/D_s$ is also observed for an elliptical pattern trajectory (see Fig. 5 down-part) for which the γ coefficient has been estimated to be approximately 1.2. The degree of invariance of γ has not been established. In particular, it may depend on the type of performance (strength, speed, etc.).

6 Discussion

This study has dealt with an analysis-synthesis method that forwards the discretization of end-effector trajectories captured from human motion and provides the command input of a synthesis model controlling a hand-arm dynamical system. Our analysis method uses a Dynamic Programming Piecewise Linear Approximation (*DPPLA*) algorithm that extracts a sequence of multi-dimensional targets from the end-effector trajectory. These targets can be used as command input of a sensory motor dynamical system. The *DPPLA* algorithm automatically computes an optimal number of target points and their respective location along the motion frames.

This discretization algorithm at the command level can be defined to identify low-dimensional sub-sets of samples not uniformly distributed in motion time series. It can be used as a segmentation method to detect points where there is a higher complexity of the trajectory. It also provides a means of reducing the data flow at the command level. Finally, by discretizing the command input it becomes possible to concatenate successive elementary motion without having to deal with the transition mechanisms.

Experiments show the effectiveness of this discrete task-based analysis/synthesis approach. The analysis of motion capture data is conducted on a set of arbitrary end-effector trajectories. The compression rate can be fixed to a high level, while maintaining satisfactory simulation results. The discretized data are then used as command input of our sensory motor synthesis system. The animation is achieved on a hand-arm mechanical system with seven degrees of freedom, associated to a control policy based on a non-parametrical learning method. In the context of motion synthesis using a data-driven approach, it might be interesting to exploit discrete information that replaces motion capture data, without reducing the quality of the animation.

This discretization mechanism is not only a mathematical tool which aims to reduce the data flow at the task-level. Indeed, it can be pointed out that the adaptive sampling is correlated to invariant laws of movement. The *DPPLA* algorithm seems to exhibit a power relation between sampled tangential velocity and the inverse of the sampled density function. The properties of this relationship, in particular the degree of invariance of the power parameter need to be further explored. Nevertheless, preliminary results tend to highlight the pertinence of the discrete patterns extracted from the *DPPLA* algorithm and in particular the link between the targets distribution along the trajectory and the curvature. In order to prove the generalization and the robustness of this law related to the *DPPLA* analysis method, more systematic tests should be conducted with various motion patterns performed by several subjects.

References

1. Vercher J.L.: Perception and synthesis of biologically plausible motion: from human physiology to virtual reality, Proceedings of Gesture Workshop 2005, (2005)
2. Kawato, M., Maeda Y., Uno, Y., Suzuki R.: Trajectory Formation of Arm Movement by Cascade Neural Network Model Based on Minimum Torque Criterion. Biological Cybernetics, Vol. 62. (1990) 275-288
3. Bullock, D., Grossberg, S., Guenther, F.H.: A Self-Organizing Neural Model of Motor Equivalent Reaching and Tool Use by a Multijoint Arm. Journal of Cognitive Neuroscience, Vol. 54. (1993) 408-435
4. Barbic J., Safonova A., Pan J.Y., Faloutsos C., Hodgins J., Pollard N.: Segmenting Motion Capture Data into Distinct Behaviors. In Proceedings of Graphics Interface 2004, (2004) 185–194
5. Boukir S., Chenevière F.: Compression and recognition of dance gestures using a deformable model, Pattern Analysis and Applications (PAA) Journal, Springer-Verlag, Vol. 7, No 3, (2004) 308-316
6. D. Chi, M. Costa, L. Zhao, and N. Badler: The EMOTE model for Effort and Shape, ACM SIGGRAPH '00, New Orleans, LA, (2000) 173-182
7. Gibet S., Marteau P.F.: A Self-Organized Model for the Control, Planning and Learning of Nonlinear Multi-Dimensional Systems Using a Sensory Feedback, Journal of Applied Intelligence, Vol. 4. (1994) 337-349
8. Lebourque T., Gibet S.: A complete system for the specification and the generation of sign language gestures. Lecture Notes in Artificial Intelligence, Lecture Notes in Artificial Intelligence 1739, in Gesture-Based Communication in Human-Computer Interaction, A. Braffort, R. Gherbi, S. Gibet, J. Richardson, D. Teil Eds., Springer-Verlag, Berlin Heidelberg (1999)
9. Gibet S., Lebourque T., Marteau P.F.: High level Specification and Animation of Communicative Gestures, Journal of Visual Languages and Computing, Vol. 12. (2001) 657-687
10. Julliard F., Gibet S., RML: A specialized Parallel Language for 3D Motion Control Specification., In International Applied Informatics Conference, Parallel and Distributed Processing Symposium, Innsbruck, Austria, (2001) 39-45
11. Gibet S., Marteau P.F., Julliard F.: Models with Biological Relevance to Control Anthropomorphic Limbs : A Survey. Lecture Notes in Artificial Intelligence. Lecture Notes in Artificial Intelligence 2298, Ipke Wachsmuth and Martin Fröhlich Eds., Springer Verlag, Berlin Heidelberg (2002) 105-119
12. Gibet S., Marteau P.F.: Expressive Gesture Animation Based on Non Parametric Learning of Sensory-Motor Models. CASA 2003, Computer Animation and Social Agents (2003)
13. Perez J.C., Vidal E.: Optimum polygonal approximation of digitized curves, *Pattern Recognition Letters*, Vol. 15. (1994) 743-750

14. Goodrich M.T.: Efficient piecewise-linear function approximation using the uniform metric. Proceedings of the tenth annual symposium on Computational geometry Stony Brook, New York, United States, (1994) 322 – 331
15. Agarwal P.K., Har-Peled S., Mustafa N.H., Wang Y.: Near-Linear Time Approximation Algorithms for Curve Simplification Proceedings of the 10th Annual European Symposium on Algorithms (2002)
16. Bellman R.: Dynamic Programming. Princeton University Press, Princeton, NJ (1957)
17. Lacquaniti, F., Terzuolo, C., Viviani, P.: The law relating the kinematic and figural aspects of drawing movements. Acta Psychologica, Vol. 54. (1983) 115-130